

記号のレイアウトに注目した数式構造認識

正員 岡本 正行[†]非会員 東 裕之^{†*}

Mathematical Expression Recognition by the Layout of Symbols

Masayuki OKAMOTO[†], Member and Hiroyuki HIGASHI^{†*}, Nonmember

あらまし 科学技術文献等の読取りシステムを実現する上で、文字部分だけでなく表や数式も適当な方法で読み取り、機械可読形式に変換しておくことが望ましい。本論文は、印刷文書中の数式を読み取る手法について述べたものである。数式では記号の2次元的配置が重要な意味をもつため、通常の記事のように単純に1次元的に読み取ることはできない。このため、数式構造を理解する手法として、記号同士の部分的な接続を処理する「個別構造処理」と呼ぶボトムアップ的処理と、数式の大まかな構造を処理する「基本構造処理」と呼ぶトップダウン的処理を組み合わせた手法を提案している。本手法により認識された数式構造は木構造として表現され、 \TeX 等のフォーマッターによりもとの数式が再現可能となっている。実験では、いくつかの論文誌から多様な構造をもつ数式の画像を切り出し、認識実験を行って本手法の有効性を確認している。

キーワード 数式認識, 文書解析, 文書認識

1. ま え が き

学術論文等の印刷文書を電子ファイル化するための文書認識システムでは、数式の部分は「2次元方向の複雑なレイアウトをもつ」、「文字のサイズがそろっていない」等の理由のために、図表等と同様に画像データのまま取り扱われることが多い。しかし、数式を画像データとして扱うことは、電子ファイル化の主目的である検索や保存などのデータ管理において、あまり有効な方法とは言えない。本研究は、学術論文等の全文読取りシステムを開発する上で重要となる数式の構造認識を行い、計算機で取扱いやすい形式で表現するための手法について述べている。

従来、数式の読取りを扱った論文は多くなく、構文解析を用いる手法や、記号同士の相対的な位置関係の統計的な性質と構文上の知識を用いた、極めて限定された範囲の数式を認識する手法が提案されているのみである^{(1)~(3)}。筆者らは以前、水平・垂直方向の空白領域を用いた再帰的分割による数式の構造認識手法を提案した^{(4)~(6)}。この手法は処理が単純で高速に実

行できる利点があるが、分割の初期段階で必ずしも適切な分割が行われず、誤った構造が抽出される場合があった。

本研究では、数式の表現において極めて基本的な特徴であると考えられる「大きな記号ほど構造を表現する役割が大きい」、「大きな空白ほど領域を分割する働きが強い」ことに注目した直感的な手法によって、数式の構造を取り出している。例えば、連分数のような分数式の構造を考えてみると、最も横幅の大きな分数の横棒を中心として領域が上下に分割され、更にその分割された各領域でも同様の分割が可能となっている。また数式の横方向の分割においても、大きな記号や空白領域は、構文上その場所で数式を左右に分割する役割を果たす可能性が高い。このような大局的な構造をトップダウン的に取り出す処理を、本手法では「基本構造処理」と呼んでいる。しかしながら、この処理だけでは斜め方向への接続関係をもつ添字式等の構造には対処できないため、あらかじめ添字式と根号式を処理しておくための「個別構造処理」と呼ぶボトムアップ的処理と組み合わせることにより、多様な数式構造の処理を可能にしている。

認識された数式構造は、計算機内部で木構造によって表現される。この木構造は必ずしも数式の意味を正しく表現するものではないが、数式構造と1対1対

[†] 信州大学工学部情報工学科, 長野市

Faculty of Engineering, Shinshu University, Nagano-shi, 380 Japan

* 現在, アンリツ株式会社

応するように作成されるため、適当なフォーマッター（本手法では $\text{T}_\text{E}\text{X}$ ）によって自動的に再現することが可能となっている。

2. 対象とする数式の範囲

数式は、極端に言えば個人が勝手に記法を決めて表記できるため、その範囲を限定することは困難である。このため、本論文では通常よく用いられている数式のフォーマッター（例えば nroff , $\text{T}_\text{E}\text{X}$ 等）を念頭において、これらのもので印刷可能な数式を認識の対象としている。しかしながら現時点では、行列（行列式）や複数行の数式を縦方向に整列させ括弧等でくくった数式では、適当な解析方法が見つからなかったため処理の対象としていない。また入力の際に接触した記号については認識ができないので、手動で正解を投入した上で本アルゴリズムを適用している。

3. 処理の概要

数式の構造認識は以下の手順に従って行われる。

(1) 前処理

- ・ 記号認識
- ・ 記号の正規化サイズ・中心位置の推定
- ・ 記号列の抽出

(2) 個別構造処理

(3) 基本構造処理

本論文では、数式中の個別記号認識後の構造認識の手法について述べる。これは記号認識と構造認識は別モジュールとして独立させることが可能であり、かつ本手法の評価において記号認識の誤りによる影響を排除するためである。以下の説明では特に断らない限り、文字、数式記号等を区別せず記号という言葉を用いる。

前処理の主な目的は、部分的な記号列の抽出である。これは関数名を一つの記号として取り扱うための文字列の抽出、個別構造処理での添字式の範囲の推定、および基本構造処理での数式の水平方向分割のための適当な空白領域選択のために用いられる。

個別構造処理は、数式中に現れるすべての根号式と添字式をあらかじめ認識し、個々の数式ブロックとして統合しておくことにより、後の基本構造処理を容易にするためのものである。この処理は記号サイズの小さいものからボトムアップ的に行われ、入れ子構造をもつものも容易に処理できるようになっている。数式は、それぞれの数式記号に応じた独自の構造（例えば分数式は、分子、分数記号、分母が垂直方向に並んだ

もの）から構成されており、この時点で他の構造も抽出して一つのブロックにまとめることも考えられるが、以下の理由により構造抽出を下記の二つのみにとどめた。

- ・ 根号式と添字式はそれぞれ包含、斜め方向の接続関係をもち、他の構造（水平・垂直方向）と異なる。

- ・ 数式記号に応じた処理モジュールを多く用意することは構文解析手法に近づき、適用可能な数式の範囲を限定する可能性がある。

基本構造処理は、「与えられた領域を大きな記号または空白で分割し、水平・垂直方向に配置されている数式の構成要素（部分式）同士の接続関係を抽出する」ものである。ここではまず数式領域全体に対して本処理が施され、分割の結果得られた各部分式の領域に対して、基本構造処理が再帰的に適用され、領域に含まれる数式要素が一つになるまで繰り返される。この処理では、領域分割のための記号、空白帯の選択は主にそれらのサイズに依存しており、必ずしも数式記号の構文規則を反映した分割が行われるとは限らないが、前書きでも述べたとおり数式の復元には支障のない木構造が得られる。

本手法の理解を容易にするため、図1に認識例を示す。図1の(a)は原画像、(b)は記号認識を行った後の記号の外接方形を示している。図1の(c), (d)は個別構造処理の例である。図中では、斜め矢印で処理の対象を、白丸で処理が終了し一つの数式要素として統合されたことを示している。図1の(e)以降は基本構造処理の例である。(e)では、まず数式全体が分数記号で上下の部分式に分割され、(f)では点線で囲まれた分数式が更に分数記号で分割されることを示している。ここで、点線で囲まれた領域は処理の対象となった一つの部分式領域を示しており、これらの包含関係は部分式としての階層構造を表している。また(g)では、点線で囲まれた分母式が斜線部分の空白領域により左右の部分式に分割され、最終的に(h)で示す分割が行われる。以下に各処理の詳細を述べる。

4. 前 処 理

前処理では、7.の認識実験で述べる記号認識の後、各記号の正規化サイズ・中央位置の推定、部分的な記号列の抽出を行っている。また以下で述べる記号同士のサイズ、中央位置の同等性の判定は、添字式の処理においても用いられる。

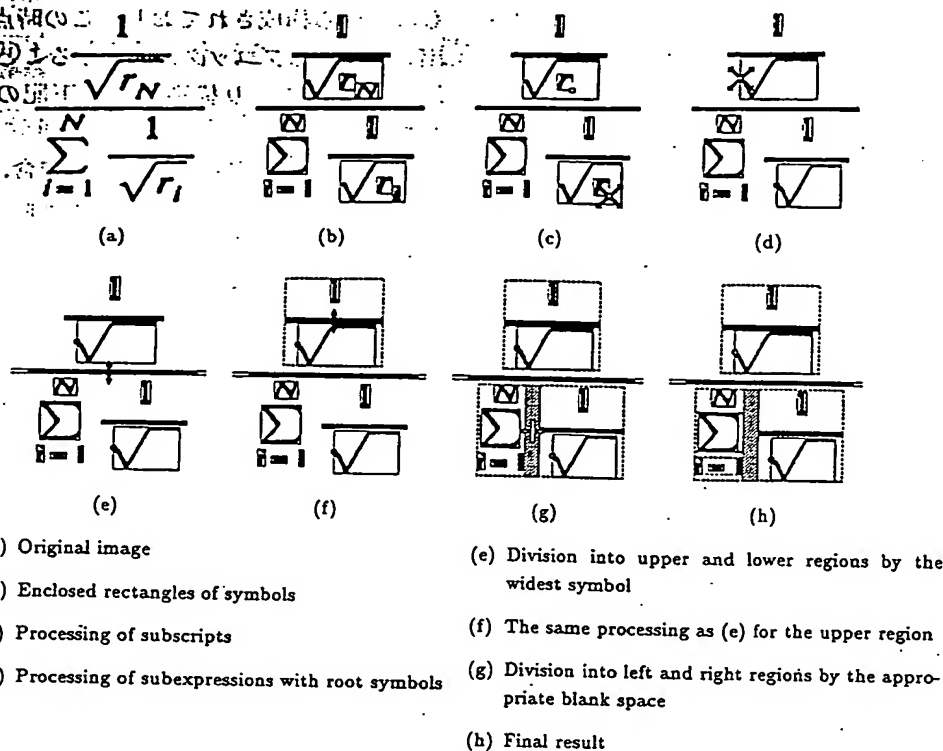


図1 認識処理の例

Fig. 1 An example of recognition process.

4.1 正規化サイズ・中央位置の推定

正規化サイズは、フォントサイズの等しいすべての記号に対して、記号を囲む最小方形の縦幅がなるべく等しくなるように調整したものである。また正規化中央位置の推定も同様に、フォントサイズの等しいすべての記号に対して、アセンディング、ディセンディング記号にかかわらず、最小方形の中央位置が等しくなるように調整するためである。このため、いくつかのスタイルの異なる同サイズのTeXフォント（ローマン体，斜体，太字体等で数学拡張フォントを除く）のフォントメトリックを利用して、以下のように求めている。

(1) 全記号の平均的サイズ（縦幅）つまり正規化サイズを求める。

(2) 記号ごとの平均サイズを求め、この値が類似している記号同士をいくつかのグループに分割した後、グループごとに正規化サイズと等しくするための係数を求める。

(3) “-”，“=”等の横長の記号8種類については、横幅を正規化サイズとする。

正規化中央位置の推定も同様に行っているが、当然

使用されるフォントによってこれらの値は異なるため、以下で述べる記号同士のサイズ，中央位置の比較ではこのことを考慮した緩やかな判定基準を採用している。

4.2 記号列の抽出

記号列の抽出では、記号同士の正規化サイズ・中央位置の同等性を以下に述べる基準で判定し、その結果「同じ記号列に所属する」と判断されたものを同一記号列としてつなげている。

(a) サイズの判定

正規化サイズを用いて、比較する二つの記号サイズの比（大きい方の記号サイズ／小さい方の記号サイズ）を計算し、1.3未満は同等，1.3以上1.5未満は不明，その他は相違と判断する。

(b) 中央位置の判定

（正規化中央位置のずれ／大きい方の記号の正規化サイズ）を計算し、0.15未満は同等，0.15以上0.19未満は不明，その他は相違とする

(c) 同一記号列の判定

サイズと中央位置の判定結果より、両者とも同等，またはサイズが同等で中央位置が不明の場合のみ，同一記号列に属するものと判定する。このとき自分のサ

イズ以上の空白が途中にあつてはならない。

以上の (a), (b) で用いた判定のためのパラメータおよび (c) の判定条件は、いくつかの数式サンプルから実験的に最適と思われるものを推定した結果である。(c) の判定条件で中央位置の判定条件が緩やかになっている理由は、中央位置の方がフォントによる変動が大きいと思われる。この記号列の抽出では、数式によっては同一記号列としては好ましくないものが抽出される可能性がある。このため本手法では、ここで抽出された記号列は 5.2.2 の添字式の範囲の推定、および 6.3 の空白帯による数式の左右への領域分割の際に補助的に用いるのみにしている。

4.3 関数名の抽出

\sin や \lim 等の関数名は、それらを一つの記号として統合しておいた方が構造認識には都合が良い。ここでは、33 個の関数名をあらかじめ登録し、文字列の最長一致により関数名を抽出している。抽出された関数名を表す文字列は、以後の処理では一つの記号として取り扱われる。

5. 個別構造処理

個別構造処理の基本的な考え方は、「サイズの小さな記号から順に、現在着目している記号（以後、着目記号と呼ぶ）に従属する部分式の範囲を推定し、その領域に対して 6. で述べる基本構造処理を適用して、その結果得られる数式構造（部分木）を、接続関係に応じたリンク（右上、右下、左上、左下および包含関係を示すもの）⁽⁶⁾を用いて、着目記号に木構造上で接続していく」というものである。このように処理を進めることにより、入れ子構造をもつ部分式も容易に処理することができる。

個別構造処理では、着目記号が根号記号の場合とそうでない場合では、部分式の接続の仕方や探索範囲が異なるので、分けて説明を行う。

5.1 着目記号が根号記号の場合

根号記号を着目記号とする場合、部分式は累乗根を表す式と根号記号内部の式の二つである。

(1) 根号記号内部の式

根号記号内部の式の領域は、その記号の上部の横線の範囲から求められる。内部式の縦方向への範囲としては、根号記号の高さをそのまま用いることができる。この領域に対し、先ほど述べたとおり基本構造処理を適用し、得られた構造を木構造上で根号記号に接続する。

(2) 累乗根式

累乗根式の横方向の範囲は、根号記号の上部横線の始点より左方向に記号の左端までとし、縦方向へは根号記号の上端からその記号の 3 分の 1 までの範囲としている。この領域に対して (1) と同様、基本構造処理を適用し得られた構造を根号記号に接続する。この領域に記号が存在しない場合には、構造がないという印を付けている。

5.2 着目記号が根号記号以外の場合

この場合には、添字式や指数式のように、着目記号から見て右上や右下等の斜め方向の接続関係が存在する。この接続関係としては、左上、左下の接続も可能であるが、右側の 2 方向（右上・右下）の処理を主に、行い、特別な記号（C, P 等）以外では、左側の部分式について処理を行わないことにしている。

5.2.1 第 1 要素の探索

これは最も着目記号に近い添字式の第 1 要素を見つける処理である。この最初の記号を「部分式の第 1 要素」と呼ぶことにする。

第 1 要素の探索範囲は、図 2 の (a) の範囲で行われる。この範囲はあくまでも基本範囲であり、着目記号の上下を検査し（図 2 の (b)）、何か記号がある場

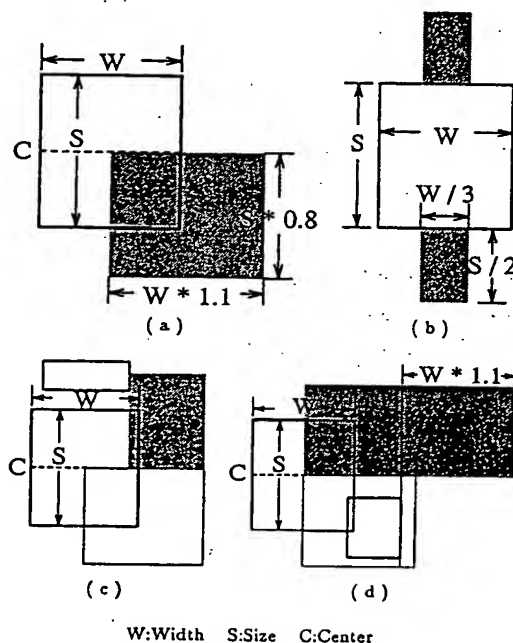


図 2 部分式の第 1 要素の探索範囲

Fig. 2 Region of search for the first symbol of a subexpression.

合には以下の処理を行う。

(1) 着目記号の上下ともに記号のある場合、またはどちらかに複数の記号がある場合は、上下限式と考えられるので添字式の処理を行わない。また着目記号が上下限式をもつ可能性のある記号 (Σ , Π 等) の場合は、上下どちらかに一つでも記号が見つければ、同様に添字式の処理は行わない。

(2) 上記以外の場合で、上下に記号が一つしか存在しなければ、その記号を含めて添字式と認識しないように、第1要素の探索開始位置をずらす (図2の(c))。

(3) 上付き部分式の第1要素を探索する際に、下付き部分式が存在していれば、その終端から更に探索範囲を図2の(d)のように拡張する。

以上のように第1要素の探索範囲を決定するが、ここに含まれる記号が第1要素となる条件は以下のものである。

- (a) 探索範囲に含まれていること
- (b) 横方向において、最も着目記号に近いこと
- (c) 着目記号と同一ライン上に存在しないこと
- (d) 着目記号よりも大きな記号でないこと
- (e) 第1要素として出現可能な記号であること

ここで同一ラインの記号とは、数式の構文上、水平方向へ同一階層レベルで並んだ記号を意味しており、本手法では正規化中心位置の比較で判断している。また(e)の条件は、着目記号と接続方向によって、添字式としてふさわしくない記号を除去するために設けたものである (例: 右下の添字式の場合 “,” で始まる添字式はふさわしくないと判断される)。もしこの時点で、第1要素が見つからなければ、その方向には部分式が存在しないものと判断する。

5.2.2 部分式の範囲の決定

上記の処理で見つかった第1要素を起点として部分式を追跡していくが、その場合に問題となるのが、部分式の範囲の決定である。一般的には、着目記号と同一ラインの記号が発見されれば、その直前までが追跡限度枠である (図3の(a))。図の(a)では、注目記号 “ Σ ” の右隣同一ラインの記号は “ a ” であり、図中の白黒反転部分が部分式の範囲とされる。ところが、「積分記号等で上下限式の部分式の範囲内に着目記号と同一ラインの記号が入り込んでいる場合」 (図3の(b)) には、部分式の記号列を分断することがなくなるまで、同一ライン記号候補をスキップしていくことにより範囲を拡張する必要がある。図3の(b)で

は、積分記号が着目記号であり、すぐ右隣の同一ライン記号は “ \sin ” 記号である。この場合、次の同一ライン記号 “ θ ” まで探索範囲が拡張される。また逆に「部分式が分数式に含まれている場合」 (図3の(c)) では、分数式の範囲によって制限する必要がある。図3の(c)の例では、この処理を行わない場合、“ k ” 以降の文字列も “ N ” の下付添字式とされる可能性がある。このため、図2の(b)の探索範囲を更に上下に2倍拡張した範囲で、注目記号がいずれかの分数式に属しているかどうかの検査を行い、もし属していれば、その分数式の範囲からはみ出ないように部分式の探索範囲を限定している。

追跡限度が決定したら、第1要素からの追跡を開始するが、そのときの追跡の条件は以下のとおりである。

- ・第1要素のサイズの60%以上のブランクが存在しないこと
- ・着目記号よりも大きな記号が存在しないこと

以上の条件が満たされている限り部分式の追跡を行い、部分式の範囲を決定する。その後、この領域を基本構造処理に渡し、得られた構造 (部分木) を右上、右下等を表すリンクで着目記号に接続する。

6. 基本構造処理

ここでは、水平・垂直方向の接続をもつ構造の処理を行う。処理の基本的な考え方は、「与えられた数式領域の分割に適した記号または空白を選択し、これによって領域分割を繰り返しながら構造の抽出を行う」というものである。これは、より大きな記号または空白ほど数式を分割する可能性が高いという視覚的な考察に基づいたものである。

ここでは以下の処理を行っている。

- (a) 記号による上下への領域分割
- (b) 記号による左右への領域分割
- (c) 空白帯による左右への領域分割

これらの処理は、上記の優先順位でまず数式領域全体に対して適用され、その結果分割された各領域 (部分式) に対して再帰的に適用される。ここで、「空白帯による上下への領域分割」がない理由は、上下への領域分割に適した空白帯の選択方法が見つからなかったためであり、これが行列式や複数行の整列式等、処理できない数式構造が存在する原因となっている。

6.1 記号による上下への領域分割

この処理は、最も横幅の広い記号によって、上下への領域分割を行う処理である。処理の手順は、以下の

$$\begin{array}{ccc} \sum_{n=1}^m a_n & \int_0^{a+l} \sin \theta & \frac{1}{2N} \sum_{k=1}^2 \end{array}$$

(a) (b) (c)

図3 部分式の追跡範囲

Fig. 3 Region of search for subexpressions.

ようになっている。

(1) 与えられた領域に含まれる記号の中で、最も横幅の広い記号を見つけ、それを着目記号とする。

(2) 着目記号の左右に同一ラインの他の記号がないか調べる。これは着目記号のみによって領域全体が上下に分割可能かどうか調べるためである。他の記号がある場合は、この処理は不可能とする。

(3) 着目記号が「分数の横棒」や「上に付く横棒」等の横長の棒状のもの（ベクトルの矢印も含む）であるときは、上と同様の理由によって与えられた領域が記号の横幅の範囲を超えている場合には、この処理は不可能とする。

(4) 領域が上下に分割可能な場合、着目記号の上部と下部の領域にそれぞれ基本構造処理を再帰的に適用し、得られた部分式を上下リンクを用いて木構造上で着目記号に接続する⁽⁶⁾。

6.1.1 実例による説明

理解を容易にするため、実例を示して上記説明の補足をする。図4の場合では、着目記号は右側の分数の横棒であるが、左側に「 Σ 」があり、このままでは領域全体を着目記号のみで上下に分割できない。このような領域は(c)の空白帯による分割の後、つまり、図の反転部分が削られた領域が作られた後、分数の横棒による上下への分割が可能となる。この上下への分割の結果、分子・分母の領域はそれぞれ基本構造処理に渡され、着目記号である分数の横棒の各部分式として接続される。

6.2 記号による左右への領域分割

この処理は、最も縦幅の広い記号によって、左右への領域分割を試みるものである。処理の手順は、以下のようにになっている。

(1) 与えられた領域に含まれる記号の中で、最も縦幅の広い記号を見つけ、それを着目記号とする。

(2) 着目記号の上下を調べ、その記号の横幅の範

$$\begin{array}{c} N-1 \\ \Sigma \\ i=1 \end{array} \quad \frac{1}{\sqrt{r_i}}$$

図4 記号による上下領域への分割の説明

Fig. 4 Explanation of division into upper and lower regions by a symbol.

囲に他の記号があれば、この処理は不可能とする。これも上記6.1の(2)と同様の理由で、着目記号のみで与えられた領域全体を左右に分割するためである。またこの検査では、記号間隔が狭くて重なっている場合を考慮して、着目記号の横幅の中心部80%のみを調べている。

(3) 着目記号の左右の領域に基本構造処理を再帰的に適用し、得られた部分式を水平リンク⁽⁶⁾を用いて着目記号に接続する。

6.3 空白帯による左右への領域分割

この処理は、最も適切な空白帯によって左右への領域分割を試みるものがある。この処理まで残された領域は、階層的に同レベルのいくつかの部分式によって構成されているはずである。このような領域では、部分式間の空白部によって領域の分割を行い、個別の数式構造を取り出す必要がある。ここでは、縦方向の空白帯を利用して、与えられた領域を左右に分割する処理を行う。処理の手順は、以下のようにになっている。

(1) 与えられた領域に存在するすべての空白帯を求める。空白帯が存在しない場合、この処理は不可能とする。

(2) すべての空白帯について、以下で述べる領域

分割への適性度を求める。ここで適性度とは、以下6.3.2と6.3.3で述べる二つのペナルティ値の和とし、この値が小さいほど適性度が高いとしている。

(3) 最も適性度の高い空白帯で、与えられた領域を左右に分割すると共に、それぞれの領域に基本構造処理を再帰的に適用し、得られた構造を水平リンクで接続する。

6.3.1 分割に適切な空白帯の条件

空白帯の条件としては、「横幅のより広いものが望ましい」という条件のほかにも、「他の部分式構造を破壊しないもの」という条件が考えられる。例えば横幅が狭くても部分式構造への影響が少ない方が望ましく、また横幅の広さ自身も、他の構造へ与える影響の大小を示していると考えられるので、両者を組み合わせて評価している。しかしながらこの時点では、領域中のすべての部分式の処理が終り、既に構造が認識されているとは限らない。このため後者の条件を適用する際は、部分式として4.3で抽出した記号列を用いている。空白帯の選択は、以下に述べるペナルティを用いて行われる。

6.3.2 空白帯の横幅の評価

空白帯の横幅を求める際には多少の注意が必要である。実際の数式中では、記号の間隔が詰まっていたり、イタリック体が使われている場合等では、単純な垂直分布による計算では空白部が求められないことも多い。そこで、分数の横棒や Σ のような横幅が重要な意味をもつ記号を除いて、記号の両端10%を削った中心部80%を正味として記号間の空白部を計算をしている。空白帯の幅によるペナルティの計算は、(与えられた領域の横幅-空白帯の横幅)として、空白帯の幅が広ければ広いほど、ペナルティが小さくなるようにしている。

6.3.3 部分式構造への影響の評価

空白帯による分割の部分式構造への影響の数値的評価は難しいが、本手法では、前に述べたように記号列に注目した方法を用いて、これは、記号列については既に抽出が行われているので、数値的な評価が行いやすいためである。ここでは、以下の二つの方針により評価を行っている。

(1) 小さなサイズの記号列の方が、数式構造上深い階層に属している可能性が高いので、できるだけ大きなサイズの記号列を分断する方が適当である。この場合のサイズとは、記号列の長さのことではなく、記号列中の記号のサイズ(縦幅)のことである。

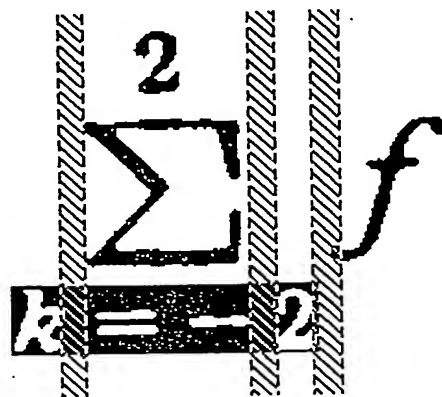


図5 空白帯による左右領域への分割の説明

Fig. 5 Explanation of division into left and right regions by a blank space.

(2) 分断される記号列は少ない方がよい。

以上の2点を考慮して、記号列分断のペナルティ計算は、(与えられた領域の高さ-分断する記号列のサイズ)とし、空白帯によって分断される記号列ごとにこの式を適用し、その総和をペナルティ値としている。

6.3.4 実例による説明

実例を示して上記説明の補足を行う。図5の場合では、横幅も縦幅も Σ が一番大きい、この記号を着目記号として上下左右ともに現段階では分割できない(分割を妨げる記号が、上下および右側に存在するからである)。そこで、空白帯による左右への領域分割が試みられる。この場合、空白帯は三つ存在する。ところが、左と中央の空白帯は、部分式の記号列を横切っている、記号列分断に対するペナルティの分だけ、右の空白帯よりも大きなペナルティをもっている。このため、領域を分割するのに最も適切な空白帯は右側のものとされ、「2」と「f」の間が分割される。

7. 認識実験および考察

本手法の有効性を確認するため、さまざまな構造をもつ数式に対して認識実験を行った。数式としては、広範囲な構造の数式を容易に印刷できる \TeX による出力のほかに、和文、英文の各2種類の論文誌から、できるだけ構造の異なるもの約50式を抽出した。なおこの式の抽出では、記号のかすれ、切れ、または接触による記号認識時の誤りを少なくするため、かなり鮮明に印刷されたもののみを用いた。数式の画像は、筆者等が開発中の文書読み取りシステム⁽⁵⁾を用いて、解像度400dpiで得られる。

本実験では、以下に述べる手法で数式中の各記号認

Original images	Recognition results reproduced by \TeX
$\lim_{n \rightarrow \infty} \frac{\log_2 K}{n - (d - 1/2) \log_2 n} \geq 1$	$\lim_{n \rightarrow \infty} \frac{\log_2 K}{n - (d - 1/2) \log_2 n} \geq 1$
$f_x^{n+1} = \bar{f}_x^n - \frac{E_x \bar{f}_x^n + E_y \bar{f}_y^n + E_t}{1 + a(E_x^2 + E_y^2)} E_x$	$f_x^{n+1} = \bar{f}_x^n - \frac{E_x \bar{f}_x^n + E_y \bar{f}_y^n + E_t}{1 + a(E_x^2 + E_y^2)} E_x$
$E = \frac{A}{2} \sum_{i=1}^N \sum_{p=1}^K \sum_{q \neq p}^K V_{pi} V_{qi} + \frac{B}{2} \sum_{i=1}^N \left(\sum_{p=1}^K V_{pi} - 1 \right)^2$	$E = \frac{A}{2} \sum_{i=1}^N \sum_{p=1}^K \sum_{q \neq p}^K V_{pi} V_{qi} + \frac{B}{2} \sum_{i=1}^N \left(\sum_{p=1}^K V_{pi} - 1 \right)^2$
$= \frac{1}{\sum_{i=1}^N \frac{1}{\sqrt{r_i}}} \left[\frac{1 - \rho_N}{\sqrt{r_N}} + \frac{\rho - \sum_{i=1}^{N-1} \frac{1 - \rho_i}{r_i}}{\sum_{i=1}^{N-1} \frac{1}{\sqrt{r_i}}} \right]$	$= \frac{1}{\sum_{i=1}^N \frac{1}{\sqrt{r_i}}} \left[\frac{1 - \rho_N}{\sqrt{r_N}} + \frac{\rho - \sum_{i=1}^{N-1} \frac{1 - \rho_i}{r_i}}{\sum_{i=1}^{N-1} \frac{1}{\sqrt{r_i}}} \right]$
$\bar{f}(r, \alpha, \beta) = \int_0^\infty w(\nu, \beta) \bar{f}\left(\frac{r}{\nu}, \alpha\right) d\nu/\nu$	$\bar{f}(r, \alpha, \beta) = \int_0^\infty w(\nu, \beta) \bar{f}\left(\frac{r}{\nu}, \alpha\right) d\nu/\nu$
(a) Correct recognition	
$= e^{a^2} (e^{a^2} - 1)$	$= e^{a^2} (e^{a^2} - 1)$
$(1 - p_i^*) N = \frac{N}{u} e^{-u^2/2}$	$(1 - p_i^*) N = \frac{N}{u} e^{-u^2/2}$
(b) Errors in recognition	

図6 数式構造認識の例

Fig. 6 Examples of mathematical expression recognition.

識を行った後、構造認識を行っているが、純粋に数式構造の認識能力のみを調べるため、記号認識部で誤って認識された記号は、構造認識に先立ってあらかじめ人が修正を行えるようになっている。

7.1 記号認識

まず8連結による黒画素連結領域のラベリングの後、各ラベルごとの記号認識が文献(7)の手法を用いて行われる。この記号認識では、水平・垂直方向等に分離した構成要素をもつ記号は、個別の記号として認識されるため統合する必要がある。記号認識において、最初から記号を一つの文字として切り出し認識しなかった理由は、数式では通常の文字サイズ、ピッチの推定による文字切出しが困難なためと、多様な数式記号に対処するためである。そこでまず分離記号の統合から述べる。

記号の構成要素の分離には、次の3種類のパターンが考えられる。

1. 上下方向への分離(「i」等)
2. 左右方向への分離(「<<」等)
3. 内外方向への分離(「@」等)

これらは、各構成要素のサイズによって決まる相対

的な距離と、要素の組合せ表による統合可能性を検査することによって、統合するかどうかの判断を行っている。上下方向分離記号の組合せ表の一部を表1に示す。この処理は上下方向に統合が可能な限り繰返される。なお表中の太い黒丸は統合不可能なことを示している。

この処理では、認識結果の厳密さが期待できないものの、例えば、「i」の下の部分に関しては、大まかなグループ分けによって、「1」と認識されたものでも適切な範囲に「i」が存在していれば、「i」に統合することになっている。

7.2 実験結果

図6は、木構造で表現される認識結果を、 \TeX に自動的に変換するフィルタを通して再現したものである。図6の(a)は正しく認識されたもの、(b)は誤認識の代表例である。なおこの図で示した認識結果では、記号認識時の誤りはすべて人手で正しく修正されている。数式認識では、文字認識実験のように系統的に大量の読取り実験を行い、認識率を定量的に評価することが難しい。このため特に認識率は求めていないが、図6の(a)から明らかなように、本手法で対象とし

表1 縦方向分離記号の組合せ表 (一部)

	上									
	.	-	~	<	>	<	>	C	∩	∪
下	.	:
	'	:
		i
	j	j
	-	.	=	≈	≤	≥	≤	≥	≤	≥
	~	.	.	≈
	=	≡	≡	≈	≤	≥	≤	≥	≤	≥
	÷
	(.

た数式 (2. 参照) はほとんどのものが正しく認識されていることがわかる。

誤認識の大半は記号同士の接触によるものであったが、本手法ではもともと処理の対象としていないため取り除いてある。その他の代表的な誤認識の例は添字式の認識である。図6の (b) の上の数式では、 α のべき乗である2が正しく認識されていない。また下の数式では、 $e^{-u^2/2}$ において u^2 の2がべき乗とならなかったため、続く / がべき乗の2とは異なる下のライン、つまり e と同列であると判断されたためである。これらの誤認識はいずれも、正規化サイズと中心位置の推定誤差によるものであるが、添字式等では記号の組合せによっては、サイズがほぼ等しく、かつ中心位置もあまり差のないものが存在し、これらの認識には別の知識が必要とされるものと思われる。

8. む す び

本論文では、記号のレイアウトに注目した数式の構造認識について述べた。ここでは、数式の構造的な特徴をよく表している大きな記号や空白領域による分割を行うトップダウン的処理と、添字式等の部分式をサイズの小さい記号から処理していくボトムアップ的処理を組み合わせた手法を提案している。本手法を、論文誌等の印刷文書中の数式に適用し、数式の構造認識がかなりの精度で可能であることを確認している。

今後の課題としては、更に大量の実験データ (数式の印刷様式が異なるものやさまざまな構造を含んだもの) による系統的評価、記号同士の接触の処理、行列式や複数行の整列式の認識、および今回の実験で誤認識を生じた添字式等の認識精度の改善等が挙げられる。

文 献

- (1) Fu K.S.: "Syntactic Method in Pattern Recognition", Academic Press, pp.245-252(1974).
- (2) Wang Z. and Faure C.: "Structural Analysis of Handwritten Mathematical Expressions", Proc. ICPR, pp.32-34 (1988).
- (3) Chou P.A.: "Recognition of Equations Using a Two-Dimensional Stochastic Context-Free Grammar", SPIE, 1199, pp.852-863(1989).
- (4) 岡本正行, 西沢 一: "数式を含む英文論文誌読取りシステムの試作と実験", 信学技報, PRU-88-158(1989-03).
- (5) Okamoto M. and Miyazawa A.: "An Experimental Implementation of a Document Recognition System for Papers Containing Mathematical Expressions", Structured Document Image Analysis, Springer-Verlag, pp.36-53(1992).
- (6) Okamoto M. and Miao B.: "Recognition of Mathematical Expressions by Using the Layout Structures of Symbols", Proc. ICDAR'91, pp.242-250(1991).
- (7) 梅田三千雄: "マルチフォント印刷漢字認識のための粗分類", 信学論 (D), J62-D, 11, pp.758-765(1979-11).

(平成6年6月17日受付, 10月24日再受付)



岡本 正行

昭46山梨大・工・電気卒。昭51東工大大学院博士課程了。同年信州大・工・情報工学科講師。現在、同大助教授。工博。形式言語、検索アルゴリズム、文書画像処理の研究に従事。情報処理学会、ACM、IEEE各会員。



東 裕之

平4信州大・工・情報卒。平6同大大学院修士課程了。同年アンリツ(株)入社。

THIS PAGE BLANK (USPTO)